

Application No: 10/761,732

Office Action Dated: June 30, 2006

Response to Office Action Dated: November 30, 2006

REMARKS

Prior to the present Amendment, claims 1-33 were pending in the application. Claims 6, 23 and 30 are cancelled herein. Claims 1, 2, 22 and 25 have been amended. Thus, claims 1-5, 7-22, 24-29 and 31-33 remain for consideration.

Claims 1-33 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,604,110 to Savage et al. (hereinafter "Savage") and U.S. Patent No. 7,065,541 to Gupta et al. (hereinafter "Gupta"). The rejection is traversed with respect to the claims as amended herein and reconsideration is respectfully requested.

Savage is directed to a method and apparatus for analyzing the relations, attributes, and rows or records of data stored within a source database to determine its metadata. The apparatus analyzes the source database in an elemental sequence that is defined by the logical and structural model of a data repository having defined entity relationships, so as to provide a comprehensive metadata model of the source data. The detailed information provided by the metadata is sufficient to provide a generic description of the source data that can be used to generate program code for a plurality of enterprise data management (EDM) applications. (Savage, col. 4, ll. 56-67). Savage further discloses that the data repository architecture supports the manual entry of migration specifications including mapping the data migration from the source database into a new target database by defining the transformations of the source data required to migrate data from the original source tables into new target tables. (Savage, col. 22, ll. 55-65). Thus, the migration specification can be utilized to move the original source data into the target database by generating an ETL job based on the specification. (Savage, col. 23, ll. 8-10).

Savage further recites:

The data repository architecture also supports the generation of many types of computing languages and other content in different formats. Often, this generated material is for use by third-party products, via some kind of import or transfer/communication mechanism. Therefore, information may be needed in order to generate the appropriate syntax/format, or to determine the correct import/transfer/communication protocol. This information might be dependent upon the characteristics and capabilities of a third-party product, or upon an applicable independent standard, such as ANSI SQL 92.
(Savage, col. 23, ll. 16-27).

Integration with such a third party tool may take a variety of forms, including but not limited to: (i) transfer of metadata between the repository and the ETLTool; or (ii) creation of a "job" in a form supported by the ETLTool, which is then made known to the tool through any supported transfer/communication mechanism; or (iii) communication to and/or from the ETLTool using a supported communication protocol/architecture.
(Savage, col. 24, ll. 8-16).

Any such integration activity may require that certain information about the third-party product be known.
(Savage, col. 24, ll. 17-18).

Thus, Savage discloses a system for the generation of executable software code for other third party systems to use for various data processing tasks which can include database migration. In the Savage system, metadata must be created, which may require user interaction throughout the process. Once the metadata has been created, it is used, in a separate step, to generate code, which is then used by a third party system for performing a task.

Therefore, the Savage system requires substantial development effort in order to transfer a database from one location to another and does not provide a client-server system for the transfer of a source database to the client across various database types, vendors and operating systems without development effort.

Clearly, the Savage system requires persons skilled in database art and the use of ETL tools to carry out a database transfer. Further, each database to be transferred must be handled separately including creating an appropriate migration specification and generating an ETL job based on the specification.

Gupta discloses a method and system for migrating a database from a first server to a second server while continuing to provide transaction service with the database during the migration. An active database is copied to a target and updated at least one time. In one embodiment, updating occurs over decreasing time intervals. When the time intervals become sufficiently short, the transition to the target server is implemented by queuing transaction requests from the source server and then executing them on the target server. (Gupta, Abstract).

The Gupta system requires the source and target database be of the same type and provided by the same vendor including a previous database to be overwritten and/or the version and release of the RDBMS at the target is compatible with the source. (See Gupta, col. 6, ll. 10-20).

In contrast, Applicants' claim 1 as amended herein, recites a client-server system for transferring a database from a server to a client including a server having a programming interface including an interface corresponding to each of a plurality of known databases stored thereon. The server being programmed to identify and load the programming interface corresponding to the source database for accessing the source database and retrieving the metadata and at least a portion of the data and storing the retrieved data in at least one data object. Amended claim 1 further recites the client including a processor for receiving and/or downloading the programming interface corresponding to the source database and receiving the metadata and the at least one data object from the server and generating and storing

a copy of the source database using the metadata, and populating the copy of the source database with the data from the at least one data object.

Nothing in the combination of Savage and Gupta teaches or suggests a client-server system for transferring a database from a server to a client including a server having a programming interface including an interface corresponding to each of a plurality of known databases stored thereon as recited in amended claim 1. Further nothing in the combination of Savage and Gupta teach or suggest a system wherein a client automatically receives and/or downloads a programming interface corresponding to the source database as also recited in amended claim 1.

As also set forth in amended claim 1, the client-server system provides for the transfer of a source database to the client across various database types, vendors and operating systems without development effort. The combination of Savage and Gupta do not disclose or suggest the claimed system wherein databases of different types can be transferred without development effort. Clearly, as set forth in Savage, the transfer of a database requires developing a specific migration specification depending on the source and target databases and generating an ETL job based on the migration specification. Accordingly, skilled users are required to carry out a database transfer using the Savage method. The Gupta system requires a common database vendor for the source and target and requires a target database to be overwritten. (Gupta, col. 6, ll.10-20).

A primary value of the present invention is providing instant, secure and effortless ("point and click" ease of use) data communication for sharing and exchanging business data among business systems across various widely used computing and data storage systems without the need for development, human intervention, pre-knowledge of data structure or collaboration among parties

Application No: 10/761,732

Office Action Dated: June 30, 2006

Response to Office Action Dated: November 30, 2006

involved with the originating data source or data destination. Such value can be significant as can be seen by the World Wide Web ("WWW" or "Web") which provides for rapid communication for sharing and exchange of human readable documents. Prior to the Web, documents could be readily exchanged, however, the exchange was not instant and effortless. The need for instant and effortless exchange of documents was addressed by the WWW and resulting value and benefits have been dramatic.

As an example, of the need for the rapid and effortless sharing of business data, consider a large manufacturing company that must share its order information with suppliers. Often, order data must be shared with hundreds of supplier companies all over the world. Typically, it is highly desirable for the company to supply order data to its suppliers instantly using one common mechanism, as opposed to working out a specific data transfer scheme with each supplier. It may also be important that any such transferred data not be exposed directly to the publicly accessible Internet. Since business data often resides on a standard relational database system, moving business data from one database to another can provide an effective and natural means of exchanging and communicating business data. Realizing and utilizing the communication potential of database transfers is one of the key values of the present invention.

Accordingly, as set forth in claim 1, the present invention provides a client-server system for transferring a database from a server to a client across various database types, vendors and operating systems without development effort. The claimed invention provides an end-to-end, stand-alone transfer operation without user intervention nor any required user knowledge of the transfer operation nor the structure of the database being transferred. (See Application, ¶ 0002). The present

Application No: 10/761,732

Office Action Dated: June 30, 2006

Response to Office Action Dated: November 30, 2006

invention is operable instantly and effortlessly with "point and click" ease of use from a GUI as shown in Fig. 8 of the Application.

In one preferred embodiment of the present invention, a GUI is automatically downloaded, installed and run as an Applet at the client, such that a user is not required to perform any explicit installation, set-up or start-up to initiate a database transfer. The user is required only to enter a small number of parameters into the GUI including selecting a data source and to define a destination for the transferred database. (See Application, Fig. 8). The database vendor is selected at the GUI using a pull-down menu which does not require user entry of any complex vendor or version numbers related to the source database nor any knowledge of the same. Once the parameters are entered, the user simply clicks a transfer button at the GUI and the system requires no other user interaction throughout the completion of the database transfer. (See Application, ¶ 0033 - 0035). Thus, as recited in claim 1, the database transfer is completed without any development effort.

Clearly, in the Savage system, to transfer database data between a source and a target database, the system requires development efforts including developing a migration specification depending on the source and target databases and generating an ETL job based on a migration specification. Further, the migration specification generated may need to be adjusted manually. Thereafter, the target database can be created and the source data migrated to it via the generated ETL job. (See Savage, col. 23, ll. 1-16). The Gupta system requires the source and target database be of the same type and provided by the same vendor. (See Gupta, col. 6, ll. 10-20). Thus, the combination of Savage and Gupta do not disclose or suggest the claimed system wherein databases of different types can be transferred automatically without user interaction or development effort.

Further, as recited in claim 18, the present invention utilizes the WWW standard protocol and therefore operates without modification of network configurations and firewalls and allows the utilization of existing WWW features such as encrypted data channels (SSL) and automatically down loadable and installable software modules (Applets) and standardized browser clients. (See Application, ¶ 0012 - 0014). Additionally, the present invention does not require that database systems involved in transferring data be directly exposed to the Internet. (Application, ¶ 0002).

Accordingly, in contrast to Savage and Gupta, the present invention system for transferring a database from a server to a client requires only minimal user time and effort and provides for the transfer of a source database to the client across various database types, vendors and operating systems without development effort. An indication of the value of minimizing the user effort required for database transfer that is provided with the present invention and distinguishes the same over the combination of Savage and Gupta can be seen in the previous example of a large manufacturing company sharing order data with a large number of suppliers. Using the present invention, a specific transfer scheme for each supplier does not need to be developed and a database can be transferred to each of a hundred or more suppliers within minutes. Whereas, the Gupta system would require at least the standardization of a database vendor for all of the suppliers, which presents an unacceptable solution. Savage, on the other hand, would require set-up and integration of one or more third party ETL tools for each supplier and would likely take weeks or months for the data sharing to begin between all of the suppliers. Further the Savage system would require ongoing effort by users skilled in database

and ETL art to maintain such a complex data transfer system which would obviously include the associated costs in labor and time.

To establish a prima facie case of obviousness for a claimed invention, all of the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Here, as set forth above, nothing in the combination of the Savage and Gupta references teaches a client-server system for transferring a database from a server to a client including a server having an interface corresponding to each of a plurality of known databases stored thereon wherein the client receives and/or downloads the interface corresponding to the source database such that a source database can be transferred to the client across various database types, vendors and operating systems without development effort. Because the combination of Savage and Gupta does not teach or suggest the above-identified features, it cannot be maintained that the cited combination teach or suggest each and every limitation of amended claim 1. Accordingly, for at least the above-identified reasons, amended claim 1 is not obvious under 35 U.S.C. § 103(a) over Savage in view of Gupta.

Claims 2-5 and 7-21 depend either directly or indirectly from claim 1 and thereby include all of the limitations of claim 1 and also include additional limitations. Since, claim 1 is not obvious under 35 U.S.C. § 103(a) over the combination of Savage and Gupta for at least the above-identified reasons, dependent claims 2-5 and 7-21 are also not obvious over the combination of Savage and Gupta.

Independent Claim 22 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Savage in view of Gupta.

Claim 22 as amended herein recites a data access application for use in transferring a database from a server to a client, the database having data stored therein and associated metadata identifying a structure and at least one field of the database. Claim 22 further recites the data access application includes an executable program for loading an interface corresponding to the database, the interface being selected from a plurality of interfaces stored on the server for use with various types of databases; retrieving from the database, the metadata and at least a portion of the data stored therein; and converting the retrieved data to Java objects and storing the Java objects in at least one data object; and transferring the metadata and the at least one data object to the server for transfer thereof to the client. Claim 22 also recites wherein the data access application operates without development or user interaction across database types, vendors and operating systems.

Nothing in the combination of Savage and Gupta disclose a data access application for use in transferring a database from a server to a client including loading an interface corresponding to the database, the interface being selected from a plurality of interfaces stored on the server, wherein data retrieved from the database is converted to Java objects and stored in a data object for transfer to the server and thereafter to the client. Further, nothing in the combination of Savage and Gupta teaches or suggests a data access application for transferring a database from a server to a client without development effort or user interaction.

As set forth above, the Gupta system requires the source and target databases be of the same type. (Gupta, col. 6, ll. 10-20). As also set forth above, the Savage system requires development of a migration specification as well as an ETL job to carry out a database transfer. (Savage, col. 23, ll. 1-16).

Application No: 10/761,732

Office Action Dated: June 30, 2006

Response to Office Action Dated: November 30, 2006

Accordingly, for at least the above-identified reasons, amended claim 22 is not obvious under 35 U.S.C. § 103(a) over Savage in view of Gupta.

Claim 24 depends from claim 22 as is therefore also deemed patentable over the combination of Savage and Gupta for at least the above-identified reasons.

Amended claim 25 recites a method for transferring a database from a server to a client, the method including a server operating on a source database across various database types, vendors and operating systems without requiring development effort. The method of claim 25 further includes storing a programming interface corresponding to each of a plurality of known databases on the server. Thus, the method provides for seamless and automatic transfer of a database without development effort regardless of the type of database or operating system. The combination of Savage and Gupta do not teach or suggest storing a plurality of programming interfaces and loading the appropriate one corresponding to the source database such that the transfer is automatic in response to a client request for the transfer. Neither Savage nor Gupta, nor the combination thereof, teach or suggest the above-identified features of amended claim 25. Accordingly, claim 25 as amended herein is not obvious under 35 U.S.C. § 103(a) over the combination of Savage and Gupta.

Claims 26-29 and 31-33 depend from claim 25 and therefore are also deemed not obvious over the combination of Savage and Gupta for at least the reasons set forth above with respect to claim 25.

In view of the foregoing, it is respectfully submitted that pending claims 1-5, 7-22, 24-29 and 31-33 are in condition for allowance. All issues raised by the Examiner having been addressed, an early action to that effect is earnestly solicited.

Application No: 10/761,732

Office Action Dated: June 30, 2006

Response to Office Action Dated: November 30, 2006

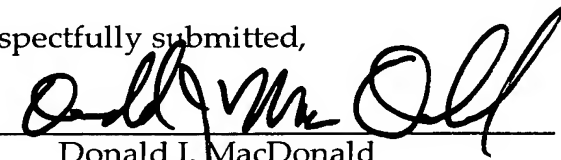
Petition for Extension of Time to Respond

Pursuant to 37 CFR 1.136(a), Applicant hereby requests a two-month extension for filing a reply to the Office Action of June 30, 2006, thereby extending the period to respond through November 30, 2006. A check in the amount of \$225 is enclosed herewith to cover the fee for the two month extension under 37CFR 1.17(a)(1).

No additional fees or deficiencies in fees are believed to be owed. However, authorization is hereby given to charge our Deposit Account No. 13-0235 in the event any such fees are owed.

Respectfully submitted,

By



Donald J. MacDonald
Registration No. 42,823
Attorney for Applicant

McCormick, Paulding & Huber LLP
CityPlace II, 185 Asylum Street
Hartford, CT 06103-3402
Tel. 860-549-5290
Fax 413-733-4543